

Testing in Ruby and Javascript

TESTING IS HARD

Q: "We all know that testing is good to do, so why is it we never really do it?"

Q: "We all know that testing is good to do, so why is it we never really do it?"

A: "Speak for yourself."

--Chris Trevarthen "TDD for iOS, Mobi Dev Day"

Why is should we test?

Why should we test?

- Assurance that things work as expected

Why else should we test?

Why else should we test?

- Refactoring -- Assurance extends through code refactoring and unexpected scope changes

Why else should we test?

- Refactoring -- Assurance extends through code refactoring and unexpected scope changes
- Environments -- test for ALL expected behaviors in a new environment

Why else should we test?

- Refactoring -- Assurance extends through code refactoring and unexpected scope changes
- Environments -- test for ALL expected behaviors in a new environment
- Updates -- to OS, library, hardware, can break working code

Why else should we test?

- Refactoring -- Assurance extends through code refactoring and unexpected scope changes
- Environments -- test for ALL expected behaviors in a new environment
- Updates -- to OS, library, hardware, can break working code
- Code portability - Bring the tests to a new platform/ language and you know what the code should do

"When I took the tests and copied them over, I had all of the necessary functionality already defined."

--Charlie Sears "Hurdles Converting a Kinect game to Mobile"

What makes it hard?

- Doing things twice
- Like learning a second language
- The hardest code to write can be the hardest code to test
- Tests take a long time to run

So, when is it not worth testing?

- Rapid Prototyping, Short turnaround
- New language that itself is a lot to learn
- Small projects with limited scope
- Projects with short lifespans

✗ 4 of 4 test failed!

✗ Rapid Prototyping, Short Turnaround

Testing can provide a roadmap for that proof of concept

Testing can help keep focus when a deadline is looming

From: Extreme Programming: A gentle introduction

*“...Extreme Programming empowers your
developers to confidently respond to changing
customer requirements, even late in the life
cycle...”*

From: Extreme Programming: A gentle introduction

*“...Unit tests are one of the corner stones of Extreme Programming (XP)... You will also create your tests first, before the code...
Unit tests are released into the code repository along with the code they test. Code without tests may not be released. If a unit test is discovered to be missing it must be created at that time....”*

```
$ rspec --format documentation
```

```
jgabel ~/Bluewater/dashboard-demos > bundle exec rspec --format documentation spec/lib/google_query_spec.rb
Run options: include {:focus=>true}

All examples were filtered out; ignoring {:focus=>true}

Google Analytics
Google Query
  with invalid credentials
    should raise argument error with invalid start or end date
    should raise argument error with invalid profile format
  with valid credentials
    should handle Ruby underscore, or Google hyphen notation
    should return a valid query
  Query Categories and Actions
    should return a valid query with only a profile parameter supplied
    should be an instance of Google Query
    should == "GoogleQuery"
  Complete Parameters
    should work with only an id supplied
    should have all required keys: ids, dimensions, metrics, start-date, and end-date,
  Stringify Parameter Array
    should respond to string
    should == "ga:eventCategory"
    should respond to array
    should == "ga:eventCategory,ga:eventAction"
  with invalid parameter and no default should respond with empty string
    should == ""
  should use default value with invalid parameter
    should == "default value"
Google OAuth Client
  with invalid credentials:
    should raise argument error with wrong key secret
    should raise authorization error with wrong email
    should raise argument error with missing key file
  with valid credentials
    saves the service_account_email
    should == "219110407277-bei2hno2tv0lpro98en7lbdcaovsd7om@developer.gserviceaccount.com"
  has a client instance
    should respond to #client

Finished in 12.52 seconds
16 examples, 0 failures
```

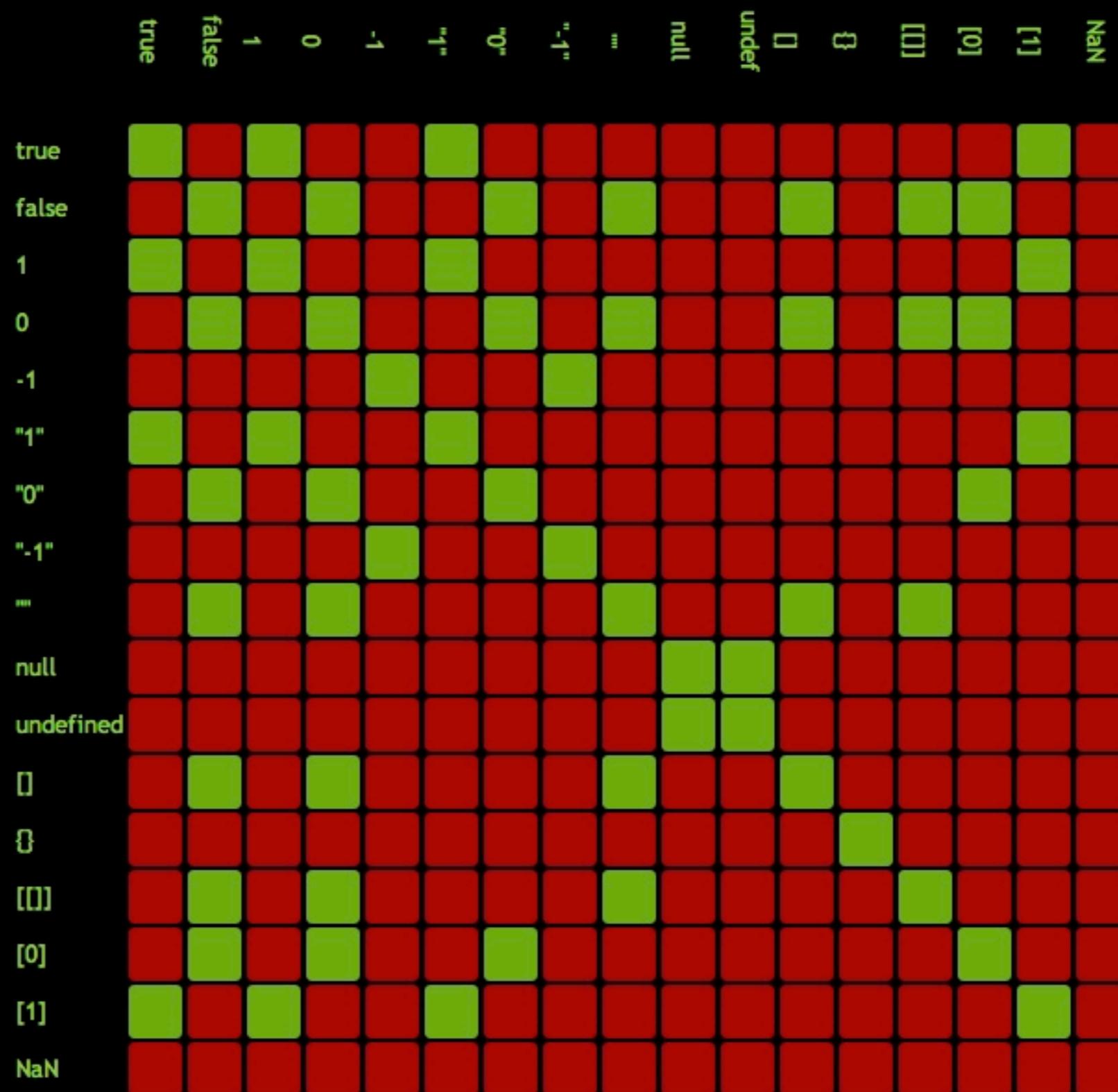
- ✖ New language that itself is a lot to learn

If the language is new to you, are you aware of all the hidden effects?

```
my_friends = [{name: "Dale", age: 52},  
              {name: "Sven", age: 26},  
              {name: "Dale", age: 13},  
              {name: "Austin", age: 37}]  
  
def everyone_has_a_unique_name? people  
  names = people.uniq! { |person| person.name }  
  people.eq? names  
end  
  
question = everyone_has_a_unique_name? my_friends
```

```
myArray = [ 0, 0, 0.0, "0", "zero",
            var zero = (function () {return 0})(),
            false, null, undefined
        ]  
  
Array.prototype.unique = function() {
    var a = this.concat();
    for(var i=0; i<a.length; ++i) {
        for(var j=i+1; j<a.length; ++j) {
            if(a[i] === a[j]){
                a.splice(j--, 1);
            }
        }
    }
    return a;
};  
  
uniqueArray = myArray.unique()
```

|||





✗ Small projects with limited scope

How many small projects continue to grow into bigger ones?

✖ Projects with short lifespans

How many projects come back from the dead?

Why do we need to test?

Why do we need to test?

WE ARE ALWAYS TESTING

We just don't always write the
tests down.

Red Green Tests:

Green == Passing
Red == Failing

Types of Test cycles:

- Green
- Red
- Green Red
- Green Red Green
- Red Green

Green

write code, see the code work

Lesson:

New code creates new functionality

Red

Write code, see the code break.
Find the code that most looks like the
problem and rewrite that code.

Lesson:

The more code you wrote, the longer it
takes to track down why it failed.

Green to Red

Submit rocking code!
Get a bug report on working code.

Lesson:
It would have been nice to find that
before someone else did.

Green Red Green

write a feature

write the passing test

comment out the feature and make sure
the test fails

Lesson:

How to write tests than relate to
functionality

Red Green Refactor

write a test

run the test and see it fail

write the feature that makes the test pass

refactor your code with confidence

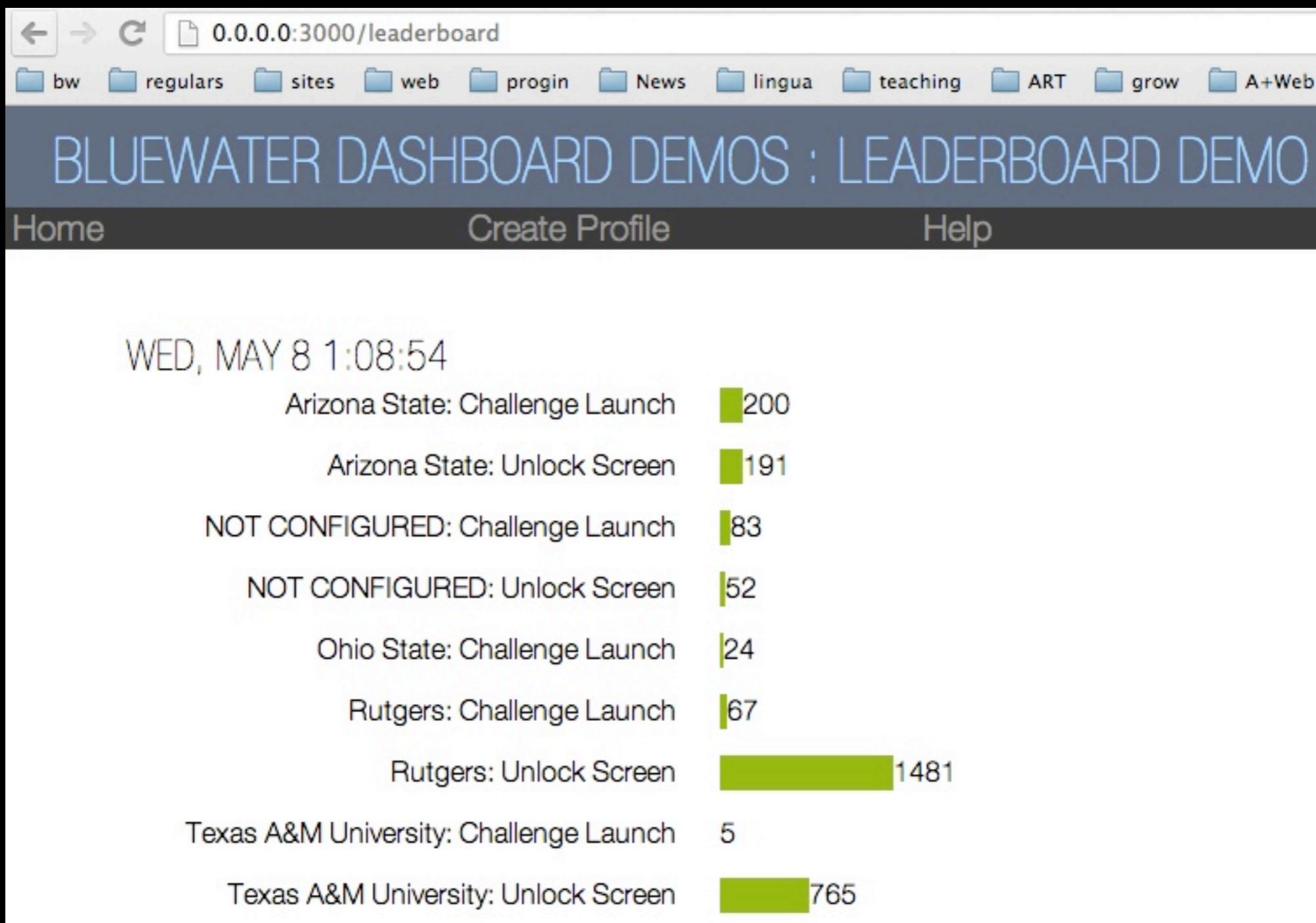
A Real World Example:

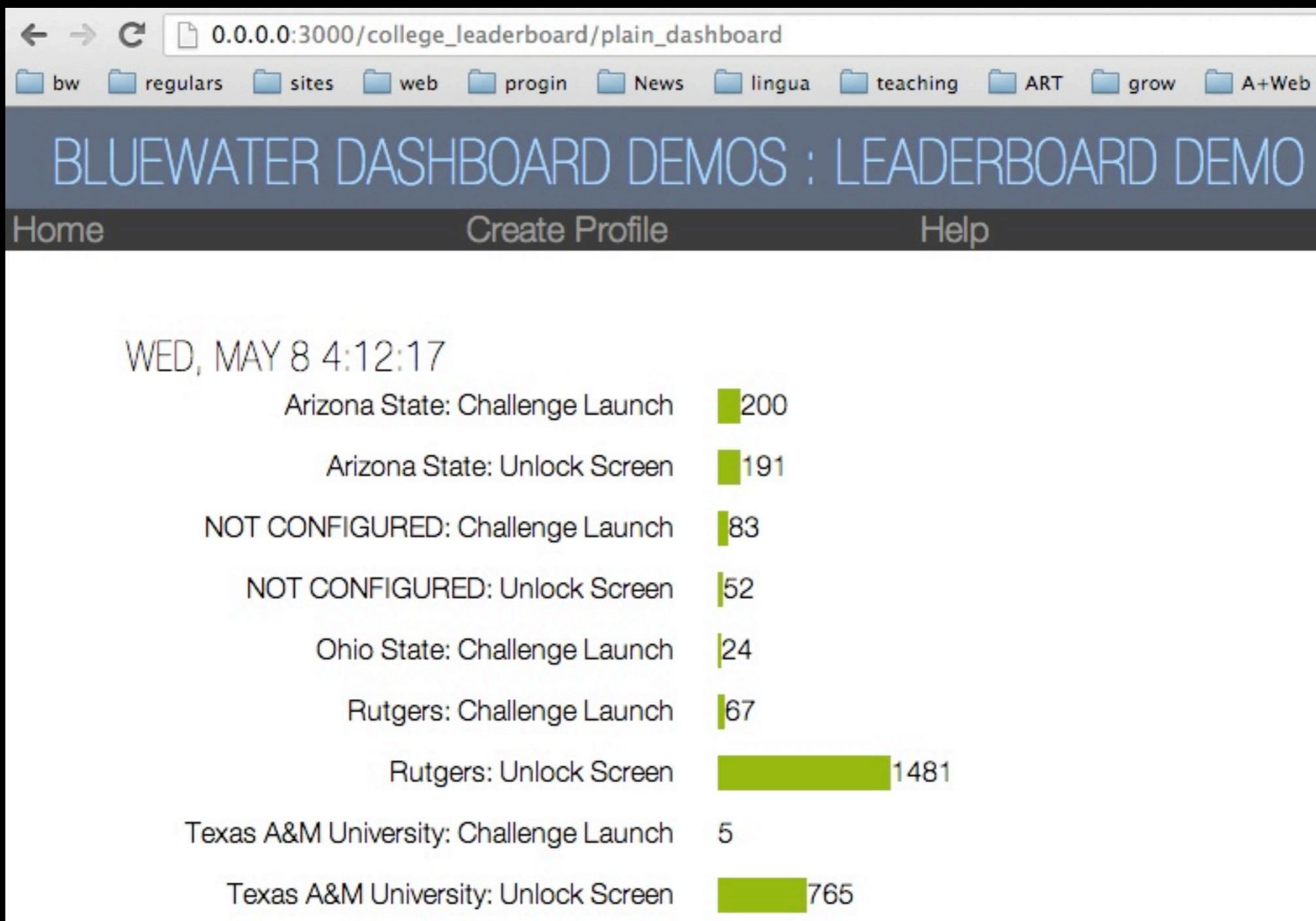
```
Finished in 8.49 seconds
69 examples, 4 failures
```

```
Failed examples:
```

```
rspec ./spec/features/college_leaderboard_spec.rb:13 # College Leaderboard College Leaderboard page
rspec ./spec/features/college_leaderboard_spec.rb:14 # College Leaderboard College Leaderboard page
rspec ./spec/support/utilities.rb:6 # College Leaderboard College Leaderboard page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # College Leaderboard College Leaderboard page it should behave like all pages
```

```
Randomized with seed 0
```





```
Coverage report generated for RSpec to /Users/jgabel/Bluewater/dashboard-demos/coverage. 137 / 327 LOC (41.9%) covered.  
/Users/jgabel/Bluewater/dashboard-demos/app/models/client.rb:15:in `<top (required)>': uninitialized constant ActiveRecord::Base (NameError)  
  from /Users/jgabel/Bluewater/dashboard-demos/spec/models/client_spec.rb:17:in `<top (required)>'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus/load_tracking.rb:50:in `load'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus/load_tracking.rb:50:in `load'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus/load_tracking.rb:43:in `load'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/rspec-core-2.13.1/lib/rspec/core/configuration.rb:819:in `block in configuration'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/rspec-core-2.13.1/lib/rspec/core/configuration.rb:819:in `each'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/rspec-core-2.13.1/lib/rspec/core/configuration.rb:819:in `load'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/rspec-core-2.13.1/lib/rspec/core/command_line.rb:22:in `run'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/rspec-core-2.13.1/lib/rspec/core/runner.rb:80:in `run'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus/rails.rb:188:in `test'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:116:in `block in command'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:103:in `fork'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:103:in `command'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:80:in `block (3 levels) in go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:80:in `fork'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:80:in `block (2 levels) in go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:73:in `each'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:73:in `block in go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:62:in `loop'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:62:in `go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:78:in `block (3 levels) in go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:78:in `fork'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:78:in `block (2 levels) in go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:73:in `each'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:73:in `block in go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:62:in `loop'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:62:in `go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:78:in `block (3 levels) in go'  
  from /Users/jgabel/.rbenv/versions/1.9.3-p392/lib/ruby/gems/1.9.1/gems/zeus-0.13.3/lib/zeus.rb:78:in `fork'
```

```
Finished in 2.19 seconds
69 examples, 23 failures
```

```
Failed examples:
```

```
rspec ./spec/support/utilities.rb:6 # Leaderboard Leaderboard page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # Leaderboard Leaderboard page it should behave like all pages
rspec ./spec/lib/google_query_spec.rb:42 # Google Analytics Google OAuth Client with valid credentials has a client instance
rspec ./spec/lib/google_query_spec.rb:47 # Google Analytics Google OAuth Client with valid credentials saves the service_account_email
rspec ./spec/lib/google_query_spec.rb:54 # Google Analytics Google OAuth Client with invalid credentials: has wrong email
rspec ./spec/lib/google_query_spec.rb:129 # Google Analytics Google Query with valid credentials should return a valid query
rspec ./spec/lib/google_query_spec.rb:141 # Google Analytics Google Query with valid credentials should query categories and actions with only a profile parameter supplied
rspec ./spec/lib/google_query_spec.rb:135 # Google Analytics Google Query with valid credentials should handle Ruby underscore, or Google hyphen notation
rspec ./spec/lib/google_query_spec.rb:86 # Google Analytics Google Query with valid credentials should be an instance of Google Query
rspec ./spec/lib/google_query_spec.rb:118 # Google Analytics Google Query with valid credentials complete parameters should work with only an id supplied should have
rspec ./spec/lib/google_query_spec.rb:93 # Google Analytics Google Query with valid credentials stringify parameter array should respond to array
rspec ./spec/lib/google_query_spec.rb:108 # Google Analytics Google Query with valid credentials stringify parameter array should respond with empty string with invalid
rspec ./spec/lib/google_query_spec.rb:103 # Google Analytics Google Query with valid credentials stringify parameter array should use default value with invalid param
rspec ./spec/lib/google_query_spec.rb:98 # Google Analytics Google Query with valid credentials stringify parameter array should respond to string
rspec ./spec/lib/google_query_spec.rb:151 # Google Analytics Google Query with invalid credentials: has invalid profile format
rspec ./spec/lib/google_query_spec.rb:157 # Google Analytics Google Query with invalid credentials: has invalid start or end date
rspec ./spec/features/static_pages_spec.rb:21 # Static pages should have the right links on the layout
rspec ./spec/support/utilities.rb:6 # Static pages Home page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # Static pages Home page it should behave like all pages
rspec ./spec/features/college_leaderboard_spec.rb:13 # College Leaderboard College Leaderboard page
rspec ./spec/features/college_leaderboard_spec.rb:14 # College Leaderboard College Leaderboard page
rspec ./spec/support/utilities.rb:6 # College Leaderboard College Leaderboard page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # College Leaderboard College Leaderboard page it should behave like all pages
```

```
Finished in 8.35 seconds
69 examples, 9 failures
```

```
Failed examples:
```

```
rspec ./spec/support/utilities.rb:6 # Leaderboard Leaderboard page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # Leaderboard Leaderboard page it should behave like all pages
rspec ./spec/features/static_pages_spec.rb:21 # Static pages should have the right links on the layout
rspec ./spec/support/utilities.rb:6 # Static pages Home page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # Static pages Home page it should behave like all pages
rspec ./spec/features/college_leaderboard_spec.rb:13 # College Leaderboard College Leaderboard page
rspec ./spec/features/college_leaderboard_spec.rb:14 # College Leaderboard College Leaderboard page
rspec ./spec/support/utilities.rb:6 # College Leaderboard College Leaderboard page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # College Leaderboard College Leaderboard page it should behave like all pages
```

```
Randomized with seed 0
```

```
/spec/support/utilities.rb

include ApplicationHelper

shared_examples_for "all pages" do
  it { should have_title(full_title("Dashboard Demos", "|", title)) }
  it { should have_selector("h1", text: full_title("Bluewater Dashboard Demos",
    ":", header_title)) }
end

/spec/features/static_pages_spec.rb

describe "Help page" do
  before { visit help_path }
  let(:title) { "Help" }
  let(:header_title) { "Help Page" }
  it_should_behave_like "all pages"
end

...
```

```
Finished in 8.49 seconds
69 examples, 4 failures
```

```
Failed examples:
```

```
rspec ./spec/features/college_leaderboard_spec.rb:13 # College Leaderboard College Leaderboard page
rspec ./spec/features/college_leaderboard_spec.rb:14 # College Leaderboard College Leaderboard page
rspec ./spec/support/utilities.rb:6 # College Leaderboard College Leaderboard page it should behave like all pages
rspec ./spec/support/utilities.rb:7 # College Leaderboard College Leaderboard page it should behave like all pages
```

```
Randomized with seed 0
```

Nomenclature!

assertion expectation test
spec behavior E2E
describe TDD BDD
testing framework test library
test runner unit testing
web driver

assertion expectation test
spec behavior E2E
describe TDD BDD
testing framework test library
test runner unit testing
web driver

assertion expectation test
spec behavior E2E
describe TDD BDD
testing framework test library
test runner unit testing
web driver

TDD: Assertion Style Tests

```
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.LastName), "FirstName is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.LastName), "LastName is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.AddressLine1), "AddressLine1 is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.AddressLine2), "AddressLine2 is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.AddressLine3), "AddressLine3 is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.City), "City is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.State), "State is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.PostalCode), "PostalCode is not empty");
Assert.IsTrue(string.IsNullOrEmpty(viewModel.Person.Country), "Country is not empty");
```

--Jeff Perrin "MVVM or What I Learned at Mobi Dev Day"

BDD: Behavior Driven Development

credited to Dan North

@collection @email @capture

Feature: Capture Guest Email

In order to capture leads customer emails should be attached to saved photos

Scenario 1: Guest photo **and** email are captured at the same **station**

Given that the **client** wants to assure email is collected

And that the **guest** experiences using the handheld device

When the **guest** approaches the activation

Then they should first have their picture taken

And be asked to enter their email

Then the email **and** photo are uploaded together to the server

Scenario 2: Guest photo is captured at one **station** **and** email at another **station**

Given that the **client** wants to capture as many photos as possible

When the **guest** approaches the activation

Then their picture is taken

And they are directed to another station to collect a token

And **when** they get to the **next** station

And select their photo from the **gallery**

Then they enter their email

assertion expectation test
spec behavior E2E
describe TDD BDD
testing framework **test library**
test runner unit testing
web driver

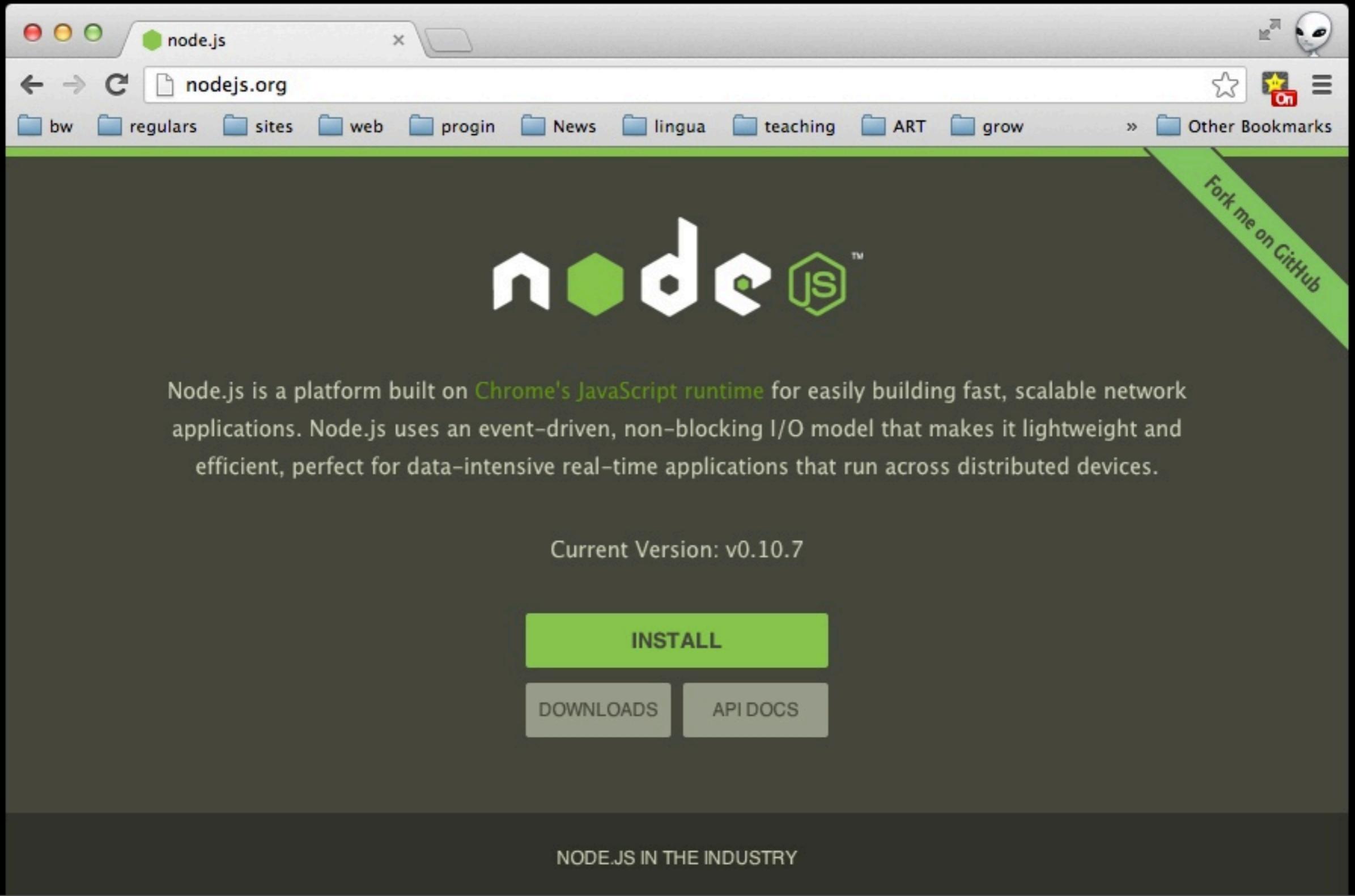
Testing in Javascript

- Browser
- Console

Node and NPM

`npm install <package> --save-dev`

Install your node package and add it to the package.json



A screenshot of a web browser window displaying the Node.js homepage. The browser has a dark theme with a light green header bar. The title bar says "node.js". The address bar shows "nodejs.org". The page content features the Node.js logo (a white 'n' and 'd' next to a green hexagon with 'e' and 's') and a green diagonal banner on the right that says "Fork me on GitHub". The main text on the page reads: "Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices." Below this, it says "Current Version: v0.10.7" and has three buttons: a large green "INSTALL" button, a smaller "DOWNLOADS" button, and a smaller "API DOCS" button. At the bottom of the page, there is a link "NODE.JS IN THE INDUSTRY".

Node and NPM

`npm install <package> --save-dev`

Install your node package and add it to the package.json

The screenshot shows a web browser window with the title bar "npm" and the URL "https://npmjs.org". The page content is the npmjs.org homepage. On the left, there's a sidebar with links: NODEJS HOME, DOWNLOAD, ABOUT, NPM REGISTRY (highlighted in red), DOCS, BLOG, COMMUNITY, LOGOS, and JOBS. The main area features the "npm" logo, a search bar labeled "Search Packages", and a user profile for "Jonathan Gabel" with a guitar icon. Below the logo, the heading "Node Packaged Modules" is displayed, followed by "Total Packages: 30 652" and download statistics: "2 248 346 downloads in the last day", "12 153 427 downloads in the last week", and "43 372 844 downloads in the last month". A red link "Patches welcome!" is present. Below that, text says "Any package can be installed by using `npm install`". Another red link "Add your programs to this index by using `npm publish`". At the bottom, two sections are shown: "Recently Updated" and "Most Depended Upon", each with a list of packages and their update times.

Recently Updated	Most Depended Upon
• 3m clean-pattern	• 2956 underscore
• 6m apic.js	• 2078 async
• 7m hubot-1stdibs	• 1836 request
• 8m grunt-jslint	• 1406 optimist
• 11m express-dispatcher	• 1395 express

Qunit unit testing framework

The screenshot shows a web browser window titled "Ask Consulting Test Suite". The URL in the address bar is "file:///localhost/Users/jgabel/Sites/bunch_web/ask_consulting/test/ask_consulting.html". The browser interface includes standard controls (back, forward, search) and a bookmark bar with items like "bw", "regulars", "sites", etc.

The main content area displays the test results:

Ask Consulting Test Suite

Module: < All Modules >

Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.93 Safari/537.36

Tests completed in 67 milliseconds.
25 assertions of 25 passed, 0 failed.

Test Description	Status	Time (ms)
1. Ask Consulting Namespace: Ask Consulting contents should be accessible (0, 1, 1)	PasSED	3 ms
2. Ask Consulting Namespace: goto method with invalid link (0, 2, 2)	PasSED	1 ms
3. Ask Consulting Namespace: goto method with valid link (0, 3, 3)	PasSED	7 ms
4. Ask Consulting Namespace: inSequenceGoto(next) method with valid link (0, 2, 2)	PasSED	4 ms
5. Ask Consulting Namespace: in sequence goto previous with valid link (0, 2, 2)	PasSED	3 ms
6. Ask Consulting Namespace: in sequence goto method with no active links (0, 2, 2)	PasSED	4 ms
7. Ask Consulting Namespace: in sequence goto method with invalid link (0, 2, 2)	PasSED	0 ms
8. DOM Manipulation: goto method should work for all content divs (0, 4, 4)	PasSED	6 ms
9. DOM Manipulation: goto method changes active content link (0, 3, 3)	PasSED	2 ms
10. DOM Manipulation: arrows clicks trigger goto in sequence (0, 4, 4)	PasSED	5 ms

PhantomJS

headless webkit

The screenshot shows the official PhantomJS website (phantomjs.org) displayed in a web browser. The page features a large 'headless webkit' logo at the top. Below it, there's a navigation bar with links like 'SOURCE CODE', 'DOCUMENTATION', 'API', 'EXAMPLES', and 'FAQ'. A prominent section highlights 'Full web stack' and 'No browser required'. A code example in a box shows how to load a webpage and exit:

```
Simple Javascript example  
console.log('Loading a web page');  
var page = require('webpage').create();  
var url = 'http://www.phantomjs.org/';  
page.open(url, function (status) {  
    //Page is loaded!  
    phantom.exit();  
});
```

Below this, there are download links for 'Download v1.9' and 'Get started'. At the bottom, there are community links: 'Community', 'Read the release notes', 'Join the mailing list', and 'Report bugs'. A large section titled 'PhantomJS is an optimal solution for' lists various applications: 'HEADLESS WEBSITE TESTING', 'SCREEN CAPTURE', 'PAGE AUTOMATION', and 'NETWORK MONITORING'. A note at the bottom states that PhantomJS is used in projects like Bootstrap, CodeMirror, Ember.js, Less.js, Modernizr, YUI3, and many more.

Qunit + Phantomjs

Used in the Grunt Build Pipeline

```
jgabel ~/Sites/bunch_web/ask_consulting > grunt
Running "jshint:gruntfile" (jshint) task
>> 1 file lint free.

Running "jshint:src" (jshint) task
>> 2 files lint free.

Running "jshint:test" (jshint) task
>> 1 file lint free.

Running "qunit:files" (qunit) task
Testing test/ask_consulting.html.....OK
>> 25 assertions passed (69ms)

Running "clean:files" (clean) task
Cleaning "dist"...OK

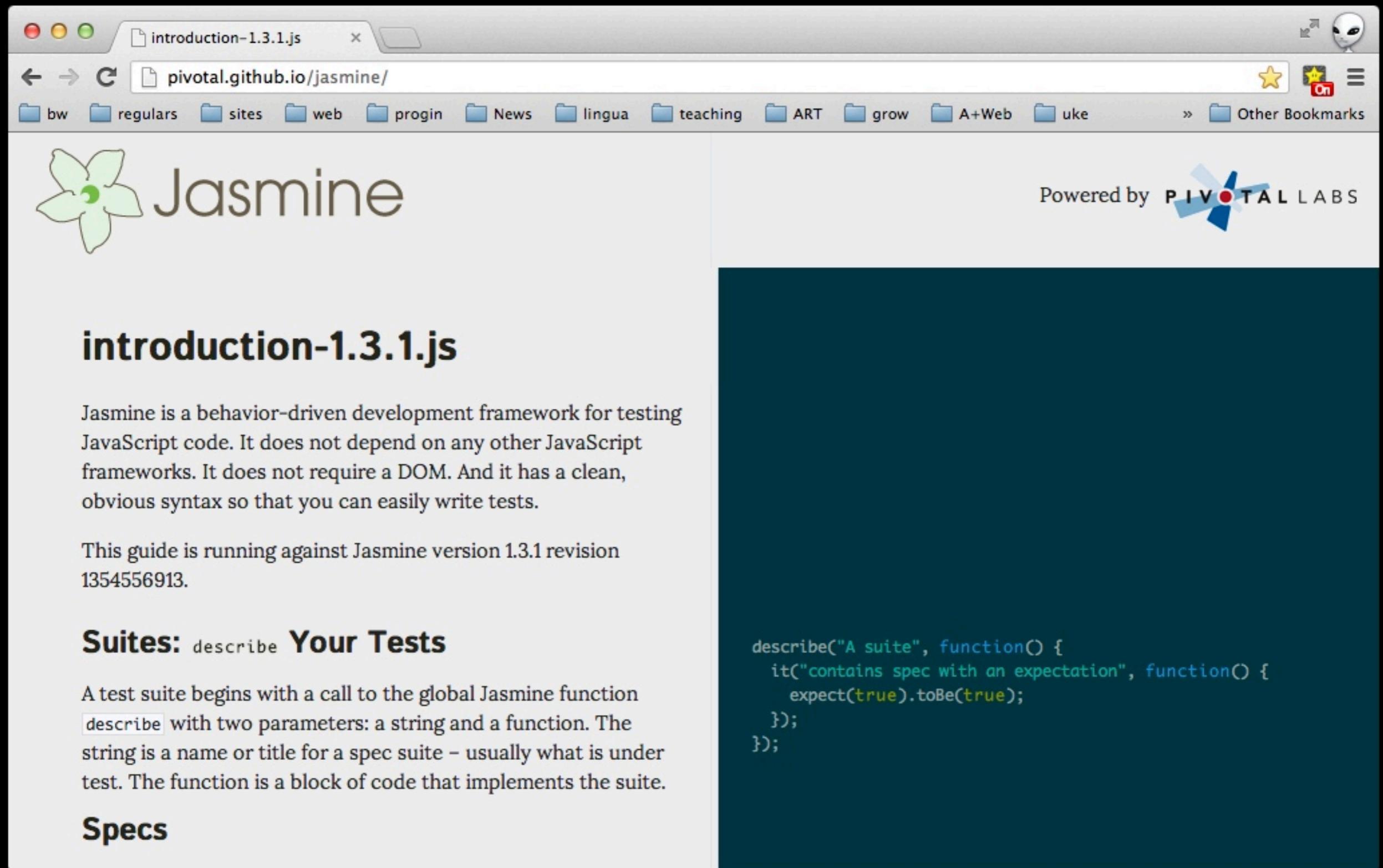
Running "concat:dist" (concat) task
File "dist/ask_consulting.js" created.

Running "uglify:dist" (uglify) task
File "dist/ask_consulting.min.js" created.
Uncompressed size: 3236 bytes.
Compressed size: 305 bytes gzipped (1377 bytes minified).

Done, without errors.
jgabel ~/Sites/bunch_web/ask_consulting > □
```

Jasmine

BDD testing framework

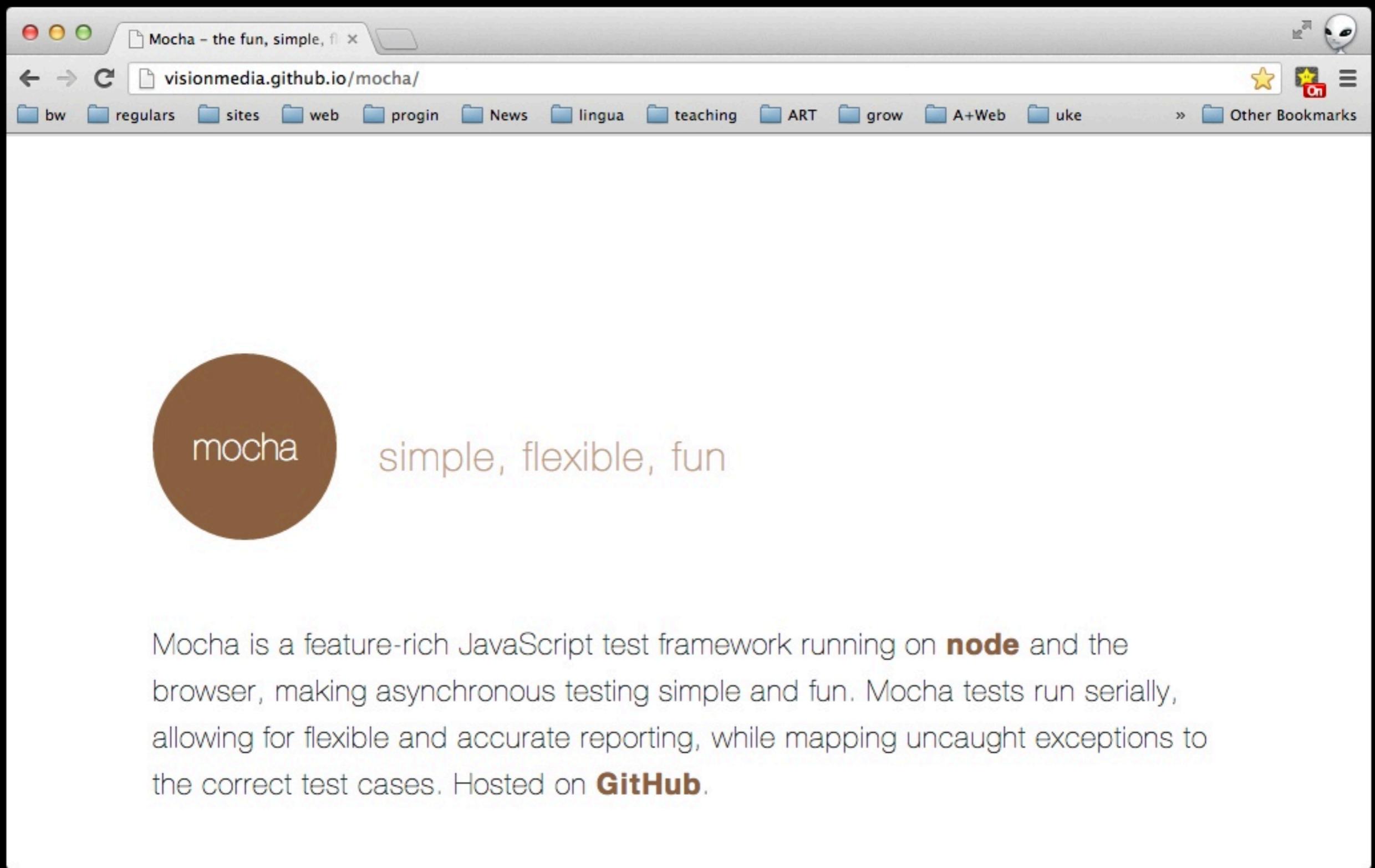


The screenshot shows a Mac OS X desktop with a web browser window open to the Jasmine homepage. The browser's title bar says "introduction-1.3.1.js". The address bar shows the URL "pivotal.github.io/jasmine/". The bookmarks bar contains links to various local folders like "bw", "regulars", "sites", etc., and a "Other Bookmarks" folder. The main content area of the browser displays the Jasmine logo (a green flower) and the word "Jasmine". Below this, the title "introduction-1.3.1.js" is prominently displayed. A paragraph explains what Jasmine is: "Jasmine is a behavior-driven development framework for testing JavaScript code. It does not depend on any other JavaScript frameworks. It does not require a DOM. And it has a clean, obvious syntax so that you can easily write tests." Another paragraph states, "This guide is running against Jasmine version 1.3.1 revision 1354556913." A section titled "Suites: describe Your Tests" discusses how a test suite begins with a call to the global Jasmine function `describe`. It explains that `describe` takes two parameters: a string and a function. The string is a name or title for a spec suite - usually what is under test. The function is a block of code that implements the suite. To the right of this text, there is a block of sample JavaScript code:

```
describe("A suite", function() {
  it("contains spec with an expectation", function() {
    expect(true).toBe(true);
  });
});
```

Mocha / Chai

Mocha: test framework

A screenshot of a web browser window displaying the Mocha test framework homepage. The browser has a dark theme with a light gray header bar. The title bar shows the window title "Mocha - the fun, simple, f..." and the URL "visionmedia.github.io/mocha/". The address bar also lists other bookmarks like "bw", "regulars", "sites", etc. The main content area features a large brown circular logo with the word "mocha" in white. To the right of the logo, the words "simple, flexible, fun" are written in a light orange font. Below this, a paragraph of text describes Mocha as a feature-rich JavaScript test framework running on Node.js and the browser, making asynchronous testing simple and fun. It mentions that Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. The text concludes by mentioning GitHub as the host.

mocha

simple, flexible, fun

Mocha is a feature-rich JavaScript test framework running on **node** and the browser, making asynchronous testing simple and fun. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. Hosted on **GitHub**.

Mocha / Chai

Chai: BDD / TDD assertion library

The screenshot shows a web browser window with the title "Home - Chai" and the URL "chaijs.com". The page content is as follows:

Chai Assertion Library

Chai is a BDD / TDD assertion library for node and the browser that can be delightfully paired with any javascript testing framework.

Download Chai v1.6.0

for Node Another platform? Browser Rails

The `chai` package is available on npm.

`$ npm install chai`

[View Node Guide](#)

[Issues](#) | [Fork on GitHub](#) | [Changelog](#) | [Google Group](#) | [Build Status](#)

Getting Started
Learn how install and use Chai through a series of guided walkthroughs.

API Documentation
Explore the BDD & TDD language specifications for all available assertions.

Plugin Directory
Extend Chai's with additional assertions and vendor integration.

Testing in Ruby / Rails

Research your gems!

`rake db:test:prepare`

Check for pending migrations and load the test schema

Test::Unit

```
require 'test/unit'

class ArrayTest < Test::Unit::TestCase

  # Fix this so that it passes the test
  def everyone_has_a_unique_name? people
    names = people.uniq { |person| person[:name] }
    people.eql? names
  end

  def test_unique

    my_friends = [{name: "Dale", age: 52},
                  {name: "Sven", age: 26},
                  {name: "Dale", age: 13},
                  {name: "Austin", age: 37}]

    question = everyone_has_a_unique_name? my_friends
    assert_equal(question, false)
  end

end
```

RSpec

**RSpec is a Behaviour-Driven Development
tool for Ruby programmers**

```
$ bundle exec rspec spec  
$ bundle exec rspec --format documentation spec
```

```
jgabel ~/Bluewater/dashboard-demos > bundle exec rspec --format documentation spec/lib/google_query_spec.rb
Run options: include {:focus=>true}

All examples were filtered out; ignoring {:focus=>true}

Google Analytics
Google Query
  with invalid credentials
    should raise argument error with invalid start or end date
    should raise argument error with invalid profile format
  with valid credentials
    should handle Ruby underscore, or Google hyphen notation
    should return a valid query
  Query Categories and Actions
    should return a valid query with only a profile parameter supplied
    should be an instance of Google Query
    should == "GoogleQuery"
  Complete Parameters
    should work with only an id supplied
    should have all required keys: ids, dimensions, metrics, start-date, and end-date,
  Stringify Parameter Array
    should respond to string
    should == "ga:eventCategory"
    should respond to array
    should == "ga:eventCategory,ga:eventAction"
  with invalid parameter and no default should respond with empty string
    should == ""
  should use default value with invalid parameter
    should == "default value"
Google OAuth Client
  with invalid credentials:
    should raise argument error with wrong key secret
    should raise authorization error with wrong email
    should raise argument error with missing key file
  with valid credentials
    saves the service_account_email
    should == "219110407277-bei2hno2tv0lpro98en7lbdcaovsd7om@developer.gserviceaccount.com"
  has a client instance
    should respond to #client

Finished in 12.52 seconds
16 examples, 0 failures
```

RSpec

```
describe "Stringify Parameter Array" do

  describe "should respond to array" do
    subject { @query.stringify_parameter_array ["eventCategory","eventAction"] }
    it { should == "ga:eventCategory,ga:eventAction" }
  end

  describe "should respond to string" do
    subject { @query.stringify_parameter_array("eventCategory") }
    it { should == "ga:eventCategory" }
  end

  describe "should use default value with invalid parameter" do
    subject { @query.stringify_parameter_array({ :invalid => "data type" }, "default value") }
    it { should == "default value" }
  end

  describe "with invalid parameter and no default should respond with empty string" do
    subject { @query.stringify_parameter_array({ :invalid => "data type" }) }
    it { should == "" }
  end

end
```

Capybara

“Capybara helps you test web applications by simulating how a real user would interact with your app.”

Capybara

```
/spec/support/utilities.rb

include ApplicationHelper

shared_examples_for "all pages" do
  it { should have_title(full_title("Dashboard Demos", "|", title)) }
  it { should have_selector("h1", text: full_title("Bluewater Dashboard
Demos", ":", header_title)) }
end

/spec/features/static_pages_spec.rb

describe "Help page" do
  before { visit help_path }
  let(:title) { "Help" }
  let(:header_title) { "Help Page" }
  it_should_behave_like "all pages"
end

it "should have the right links on the layout" do
  visit root_path
  # capybara 2: 'first' resolves ambiguous match
  first(:link, 'Help').click
  # capybara 2: 'have_title' finds invisible title text
  page.should have_title( full_title("Dashboard Demos", "|", "Help") )
  first(:link, 'Home').click
  page.should have_title( full_title("Dashboard Demos", "|", "") )
end
```

rspec-example-steps

Given/ When/ Then/ And/ But steps for
RSpec examples

JaRvaUscBriYpt

Konacha

Run mocha and chai through the asset pipeline

/Gemfile

```
group :test, :development do
  gem "konacha", "~> 2.7.0"
  gem 'selenium-webdriver'
end
```

Given app/assets/javascript/anyfile.js

spec/javascript/anyfile_spec.js

```
//= require anyfile
```

Konacha

```
$ bundle exec rake konacha:run
```

```
.....
```

```
Finished in 0.00 seconds
```

```
5 examples, 0 failed, 0 pending
```

```
$ bundle exec rake konacha:serve
```

```
Your tests are here:
```

```
http://localhost:3500/
```

```
>> Thin web server (v1.5.1 codename Straight Razor)
```

```
>> Maximum connections set to 1024
```

```
>> Listening on 0.0.0.0:3500, CTRL+C to stop
```

Konacha

A screenshot of a web browser window titled "Konacha Tests" displaying a test run. The URL is "localhost:3500". The browser's toolbar includes standard icons for back, forward, and search, along with a star icon for bookmarks and a menu icon. Below the toolbar is a bookmarks bar with various folder icons labeled "bw", "regulars", "sites", "web", "progin", "News", "lingua", "teaching", "ART", "grow", "A+Web", "uke", "Imported", and "testing". To the right of the bookmarks bar are buttons for "Other Bookmarks" and a search field. The main content area shows a list of test cases under the heading "Konacha should connect to test via the asset pipeline". Each test case has a green checkmark next to it. The test cases are:

- ✓ Should be able to access variables in assets/javascripts
- Mocha assert method
 - ✓ should work with or without Chai
- Chai Assert
 - ✓ should work within a mocha test
- Chai Expect
 - ✓ should work within a mocha test
- Chai Should
 - ✓ should work within a mocha test

In the bottom right corner of the content area, there is a circular progress bar with the text "passes: 5 failures: 0 duration: 0.03s 100%".

Travis

*“A hosted continuous integration service for
the open source community.”*

Karma (Testacular)

“On the AngularJS team, we rely on testing and we always seek better tools to make our life easier. That's why we created Karma - a test runner that fits all our needs.”

Grunt

“If I was starting the project today, I would definitely use [Yeoman](#) to easily generate templates and make life easier.” --Jeff Hooks [*Lessons Learned: A Year with a Large AngularJS Project*](#)



The screenshot shows a Mac OS X desktop with a web browser window open to the Grunt website at gruntjs.com. The browser's address bar shows the URL. The page features a large orange header with the Grunt logo (a cartoon bull's head) on the left. To the right of the logo, the word "GRUNT" is written in large, bold, dark brown capital letters, followed by "The JavaScript Task Runner" in a slightly smaller font. Above the main title, there are navigation links: "Getting Started", "Plugins", "Documentation", and "API". Below the main title, there is a yellow circular badge with the ".net AWARDS" logo and the text "VOTE NOW NOMINEE". The page content includes sections for "Latest Version" (listing v0.4.1 as stable and N/A as development), "Why use a task runner?", and "Why use Grunt?". A sidebar on the left contains a "Latest News" section and a "Bocoup" advertisement for "Optimizing Your Workflow with Grunt".

Latest Version

- Stable: v0.4.1
- Development: N/A

 [Optimizing Your Workflow with Grunt](#): a new class on air from the experts at Bocoup.

Latest News

Why use a task runner?

In one word: automation. The less work you have to do when performing repetitive tasks like minification, compilation, unit testing, linting, etc., the easier your job becomes. After you've configured it, a task runner can do most of that mundane work for you—and your team—with basically zero effort.

Why use Grunt?

The Grunt ecosystem is huge and it's growing every day. With literally hundreds of plugins to choose from, you can use Grunt to automate just about anything with a minimum of effort. If someone hasn't already built what you need, authoring and publishing your own Grunt plugin to npm is a breeze.

Spork || Zeus

```
jgabel ~/Bluewater/dashboard-demos > zeus start
Starting Zeus server
[ready] [crashed] [running] [connecting] [waiting]
boot
└── default_bundle
    ├── development_environment
    │   └── prerake
    └── test_environment
        └── test_helper

Available Commands: [waiting] [crashed] [ready]
zeus generate (alias: g)
zeus server (alias: s)
zeus destroy (alias: d)
zeus console (alias: c)
zeus dbconsole
zeus runner (alias: r)
zeus rake
zeus test (alias: rspec, testrb)
-
```

<https://github.com/josankapo/TestingIsHard>